UNITED STATES PATENT APPLICATION
FOR
ONE-WAY BROADCAST KEY DISTRIBUTION

INVENTORS:

DAVID A. LEE
a citizen of The United States, residing at
740 SW WILLOW CREEK DRIVE
BEAVERTON, OREGON 97006

MICHAEL S. RIPLEY
a citizen of The United States, residing at
1222 NE 56TH CT
HILLSBORO, OREGON 97214

PREPARED BY:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1026
(303) 740-1980

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number: EL899343385US
Date of Deposit: September 28, 2001
I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Commissioner of Patents and Trademarks, Washington, D. C. 20231

Debbie Peloquin
(Typed or printed name of person mailing paper or fee)

Debbie Peloquin
(Signature of person mailing paper or fee)

September 28, 2001
(Date signed)

ONE-WAY BROADCAST KEY DISTRIBUTION

FIELD OF THE INVENTION

[0001] The invention relates generally to the field of data encryption. More particularly, the invention relates to a one-way broadcast distribution of keys.

BACKGROUND OF THE INVENTION

[0002] Providers of digital content of various types frequently broadcast this content via various media. Examples of the type of content include music, multimedia presentations, text, television content, software, and other forms of digital data. Types of broadcasts can include various forms of over-the-air broadcasts, copper wire or fiber optic cable based network broadcasts, or even distribution of recordable media such as magnetic or optical disks. Regardless of the media used, all of these types of broadcasts are one-way distributions of content.

[0003] When distributing content in such a manner, the content provider frequently wishes to encrypt the content to prevent unauthorized persons from receiving the content. The problem is, the keys for decrypting the content must be sent to the receiver also. Frequently, this key is broadcast along with the content. Unfortunately, interception of this key then becomes relatively easy. Additionally, cracking the key to provide unauthorized access to the content, while possibly time consuming, also becomes relatively easy.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] The appended claims set forth the features of the invention with particularity. The invention, together with its advantages, may be best understood from the following detailed description taken in conjunction with the accompanying drawings of which:

[0005] **Figure 1** is a block diagram illustrating a high-level view of a system for one-way broadcast key distribution according to one embodiment of the present invention;

[0006] **Figure 2** is a block diagram of a key distribution center system according to one embodiment of the present invention;

[0007] **Figure 3** is a block diagram of a receiver system according to one embodiment of the present invention;

[0008] **Figure 4** is a flowchart illustrating a high-level view of one-way broadcast key distribution according to one embodiment of the present invention;

[0009] **Figure 5** is a flowchart illustrating key distribution center processing according to one embodiment of the present invention;

[0010] **Figure 6** is a flowchart illustrating an update key generation process according to one embodiment of the present invention;

[0011] **Figure 7** is a flowchart illustrating receiver processing according to one embodiment of the present invention; and

[0012] **Figure 8** is a flowchart illustrating a list parsing process according to one embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0013] A method and apparatus are described for a one-way broadcast distribution of keys for decrypting encrypted broadcast content. According to one embodiment of the present invention, a method and apparatus are described for generating a list of update keys on a content provider system based on a table of secret keys associated with a plurality of content receivers. The list of update keys is generated in a manner to allow valid receivers to recover a valid content key while invalid receivers recover an invalid content key. The list of update keys are used to generate a multiple nested list of decryption patterns that is broadcast to all receivers. The receivers then recover an appropriate set of update keys for each receiver from the multiple nested list of decryption patterns so that the final key recovered in the set of update keys is a content key.

[0014] In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without some of these specific details. In other instances, well-known structures and devices are shown in block diagram form.

[0015] The present invention includes various steps, which will be described below. The steps of the present invention may be performed by hardware components or may be embodied in machine-executable instructions, which may be used to cause a general-purpose or special-purpose processor or logic circuits programmed with the instructions to perform the steps. Alternatively, the steps may be performed by a combination of hardware and software.

[0016] The present invention may be provided as a computer program product which may include a machine-readable medium having stored thereon instructions which may be used to program a computer (or other electronic devices) to perform a process according to the present invention. The machine-readable medium may include, but is not limited to, floppy diskettes, optical disks, CD-ROMs, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, magnetic or optical cards, flash memory, or other type of media / machine-readable medium suitable for storing electronic instructions. Moreover, the present invention may also be downloaded as a computer program product, wherein the program may be transferred from a remote computer to a requesting computer by way of data signals embodied in a carrier wave or other propagation medium via a communication link (e.g., a modem or network connection).

[0017] Importantly, while embodiments of the present invention will be described with reference to a broadcast of content such an over-the-air broadcast or network broadcast, the method and apparatus described herein are equally applicable to other forms of content distribution. For example, the techniques described herein are thought to be useful in connection with the distribution of content on optical or magnetic recordable media such as compact disks.

[0018] **Figure 1** is a block diagram illustrating a high-level view of a system for one-way broadcast key distribution according to one embodiment of the present invention. This system includes a Key Distribution Center (KDC) 105, a broadcaster 110, and a number if receivers 115, 120, and 125. In this example, the broadcaster 110 is intending to broadcast 140 encrypted content to the receivers 115-125. This broadcast 140 may occur over a variety of media. For example, the broadcast 140 may be an over-the-air broadcast, or may be sent over a network of copper wire or fiber optic cable. In another situation, content may be distributed on magnetic or optical recordable medium such as a compact disk. In any case, encrypted content is transferred from the broadcaster 110 to the receivers 115-125 in a one-way, one-to-many manner.

[0019] If all receivers are eligible to receiver the content, one key, known to both the broadcaster 110 and the receivers 115-125, may be used. However, not all of the receivers 115-125 may be eligible to receive the content. For example, one of the receivers may not have paid a subscription fee for a particular piece of content or may have been altered or hacked in some manner. If known to the broadcaster 110, the unauthorized or "bad" receiver may be blocked from receiving some or all content. In order to block unauthorized receivers, the broadcaster 110 identifies bad receivers and notifies 135 the KDC 105 of their identity. Alternatively, another entity, such as a licensing agent or the even the KDC 105, handles the task of identifying bad receivers. Regardless of who identifies the bad receivers, the KDC 105 maintains a table of secret keys for all receivers 115-125 in the system. The KDC then generates a list of update keys based on the table of secret keys. Details of how this list may be generated are discussed below with reference to figure 6. Generally, the list of update keys is generated in a manner to allow valid, authorized receivers to recover a valid content key while invalid, unauthorized receivers recover an invalid content key. In other words, the KDC 105 generates a chain of intermediate update keys. Depending on whether the keys are intended for a valid or invalid receiver, the chain leads to a valid or invalid content key.

[0020] The update keys are then used to generate a multiple nested list of decryption patterns that contains versions of the update keys encrypted using the secret keys assigned to each receiver. Therefore, the update keys themselves are protected through this encryption. The multiple nested list of decryption patterns is broadcast 145 from the KDC 105 to all receivers 115-125. The receivers 115-125 then recover an appropriate set of update keys for each receiver 115-125 from the multiple nested list of decryption patterns so that the final key recovered in the set of update keys is a content key. Details of how these keys may be recovered are discussed below with reference to figure 8. Additionally, the KDC sends 130 the content key to the broadcast 110 for use in encrypting content to be sent to the receivers. Alternatively, the content key may be generated by another entity. For example, the broadcaster or a third party may generate the content key as long as the key can be shared by the broadcaster and KDC. Broadcast content may be encrypted with the content key using any of a variety of well-known encryption techniques.

[0021] **Figure 2** is a block diagram of a key distribution center system according to one embodiment of the present invention. In this example, a key distribution center (KDC) system 200 receives 225, from a broadcaster, an indication of which receivers to exclude from a coming broadcast. As mentioned above, in alternative embodiments, another entity, such as a licensing agent or the even the KDC 200, handles the task of identifying bad receivers. This indication may be in the form of a list of identifiers uniquely identifying bad receivers. This indication is then used by a process 205 within the KDC system 200 that controls the inclusion or exclusion of particular receivers. This process 205 may, in some manner, indicate or flag bad receivers in the table of receiver secret keys 210. Alternatively, the receiver include/exclude control process 205 may directly influence the update key generation process 215 rather than writing an indication to the table of receiver secret keys 210.

[0022] The table of receiver secret keys 210, in the following simple example, has the dimension of 2 x 2.

$K_{0,0}$	$K_{1,0}$
$K_{0,1}$	$K_{1,1}$

[0023] Using this table, each receiver can be assigned one key from each column. The combination of these keys then uniquely identifies that receiver. For example, with this table, four possible receivers can be identified. They are the receivers identified by the combination of secret keys $K_{0,0}$ and $K_{1,0}$, $K_{0,0}$ and $K_{1,1}$, $K_{0,1}$ and $K_{1,0}$, and $K_{0,1}$ and $K_{1,1}$. Of course, larger tables would be used in actual implementations. Additionally, various numbers of rows and columns may be used depending on the particular application.

[0024] The content key generation process 220 generates a content key to be used in encrypting and decrypting a future broadcast of content. The key is sent 230 to a broadcaster for encrypting the content and is used by the update key generation process 215. This process 220 can randomly generate a key suitable for use with whatever method is being used to encrypt the content.

[0025] The update key generation process 215 generates a list of update keys based on the table of secret keys 210. Details of how this list may be generated are discussed below with reference to figure 6. Generally, the list of update keys is generated in a manner to allow valid, authorized receivers to recover a valid content key while invalid, unauthorized receivers recover an invalid content key. In other words, the KDC 105 generates a chain of intermediate update keys. Depending on whether the keys are intended for a valid or invalid receiver, the chain leads to a valid or invalid content key.

[0026] The update keys are then used to generate a multiple nested list of decryption patterns that contains versions of the update keys encrypted using the secret keys assigned to each receiver. Therefore, the update keys themselves are protected through this encryption. The multiple nested list of decryption patterns is broadcast from the KDC 105 to all receivers 235.

[0027] **Figure 3** is a block diagram of a receiver system according to one embodiment of the present invention. In this example, a receiver 300 receives 325 from the KDC a multiple nested list of decryption patterns in which the update keys have been encrypted. A list parsing/key recovery process 305 then reads the list and recovers the update keys intended for this receiver 300. Details of this process 305 will be discussed below with reference to figure 8. Generally, the result of this process 305 is a valid content key 310 if the receiver is authorized to receive content. If the receiver 300 is not authorized to receive content, the result of the list parsing/key recovery process is an invalid content key.

[0028] A broadcast receiver 315 will receive 330 encrypted content from the broadcaster. Details of this receiver 315 are well-known to those skilled in the art. The receiver 315 may be any type of receiver suitable for receiving transmissions from the broadcast over the applicable medium.

[0029] The content decryption process 320 uses the content key 310 to decrypt the content received by the broadcast receiver 315. Assuming a valid content key 310, the content decryption process results in usable content provided 335 to a viewer or end user. As mentioned above with regard to the encryption process of the broadcaster, the decryption process can be the complement of any of the well-known encryption methods that may be used by the broadcaster.

[0030] **Figure 4** is a flowchart illustrating a high-level view of one-way broadcast key distribution according to one embodiment of the present invention. Initially, at processing block 405, a list of update keys are generated for each receiver from the table of secret keys stored at the KDC. Details of this process will be described below with reference to figure 6. The KDC then, at processing block 407 generates a multiple nested list of decryption patterns that contains versions of the update keys encrypted using the secret keys assigned to each receiver. Next, the KDC broadcasts the multiple nested list of decryption patterns to all receivers at processing block 410. The receivers then, at processing block 415, recover a set of update keys for that receiver in order to obtain the content key. Next, at processing block 420, a broadcaster distributes content encrypted with the content key. Finally, all receivers with a valid content key can decrypt the broadcast content at processing block 425. Alternatively, encrypted content may be broadcast to all receivers prior to or concurrent with broadcast of the multiple nested list of decryption patterns if the content will be cached prior to decryption.

[0031] **Figure 5** is a flowchart illustrating key distribution center processing according to one embodiment of the present invention. First, at processing block 505, bad, unauthorized receivers are identified. As explained above, a broadcaster notifies the KDC of which receivers are authorized to receive content and which receivers are not authorized. This notification may be in the form of a list of identifiers. Next, at processing block 510, the KDC may optionally update the table of receiver secret keys. That is, the KDC may flag or otherwise mark entries in the table of secret keys that relate to unauthorized receivers. The KDC then, at processing block 515, generates a list of update keys for all receivers. Details of this process will be described below with reference to figure 6. Next, at processing block 517, the KDC generates a multiple nested list of decryption patterns that contains versions of the update keys encrypted using the secret keys assigned to each receiver. Finally, the KDC transmits the multiple nested list of decryption patterns to all receivers at processing block 520.

[0032] **Figure 6** is a flowchart illustrating an update key generation process according to one embodiment of the present invention. Basically, this process involves generating a list of data based on a table. As will be easily understood by those skilled in the art, many possible methods can used to accomplish this. The example offered here is provided to illustrate one possible method but is not intended to exclude other possibilities.

[0033] In this example, processing begins with the first column of the table. A determination of the number of entries in the list for each row of the column is made at processing block 605. This determination is based on the number of update keys generated for the previous column. For the first column, no previous update keys are used, so, only one entry per row is needed. However, in another column of the table, if three update keys were sent for a previous column, the current column uses three entries per row.

[0034] Next, at processing block 610, the number of update keys to be used for this column is determined. This determination is based on the number of possible unauthorized receivers identified by entries in this column. For example, if no unauthorized receivers are identified, only one key is needed. That is, if there are no unauthorized receivers indicated, only a good, valid update key is sent. However, if there are possible unauthorized receivers indicated, a different update key can be generated for each possible bad receiver plus the one update key for good receivers. For example, if the key combinations represented in the current column indicate two possible bad receivers, three update keys are generated, one for good receivers and one each for possible bad receivers indicated in this column.

[0035] The proper number of update keys is then generated at processing block 615. As indicated above, the keys can be generated using any of the methods that are well known in the art. In some applications, the keys may be randomly generated. The primary considerations in generating the keys are that they be compatible with the encryption method used and that they are difficult to guess.

[0036] At processing block 620, the update keys are encrypted with a key that is a combination of the previous update key, the device secret key associated with this row and column, and table location, using some reversible function such as exclusive or. A test pattern is also provided for the associated previous update key for this entry. The test pattern is a fixed pattern that is known to all receivers. The purpose of the test pattern is to enable the receivers to locate keys intended for that receiver within the list of keys. As will be explained below with reference to figure 8, the receivers parse the list, locate keys intended for this receiver at this step based on finding the expected test pattern, and decrypt the associated entries in the list.

[0037] At processing block 625, the encrypted update keys are appended to the multiple nested list of decryption patterns. At decision block 630, if more rows exist in the current column, processing returns to processing block 620. That is, keys are generated, encrypted and appended to the list for each row of the current column. At decision block 635, if more columns exist in the table, processing returns to processing block 605. That is, keys are generated for all columns in the table.

[0038] Alternatively, if there are no bad receivers to block, the first column is used. That is, only one update key is needed. Generally if any bad receivers are to be blocked, all of the columns are processed. In some embodiments, device key assignments may be grouped in some way to allow use of an intermediate number of columns to block a selected group.

[0039] **Figure 7** is a flowchart illustrating receiver processing according to one embodiment of the present invention. First, at processing block 705, the receiver receives the list of update keys from the KDC. Next, the receiver parses the list at processing block 710 to obtain the update keys for that receiver. Details of this process will be described below with reference to figure 8. Finally, at processing block 715, the receiver determines the content key from the chain of update keys by using the first update key to decrypt the second and so on until the last key in the chain represents the content key.

[0040] **Figure 8** is a flowchart illustrating a list parsing process according to one embodiment of the present invention. Basically, this process involves parsing a list of data. As will be easily understood by those skilled in the art, many possible methods can be used to accomplish this. The example offered here is provided to illustrate one possible method but is not intended to exclude other possibilities.

[0041] First, at processing block 805, the receiver reads an entry from the list. The test pattern is extracted from the decrypted data at processing block 815 by performing the compliment function of that used to combine the data at the KDC. If the test pattern extracted from the list entry matches that expected at decision block 820, the entry is decrypted using the receiver's secret keys at processing block 822 and the update key from that list entry is recorded at processing block 825 as being intended for use by this receiver. Finally, at decision block 830, processing returns to processing block 805 if more list entries are present.

[0042] To further illustrate the process described above, the following examples are provided. The examples describe a particular manner of reading a table of secret keys, generating a list of update keys and later parsing the list to recover the appropriate update keys. However, this particular method is described only as an example. Other well-known methods of performing these functions may be used.

[0043] For a first example, consider a simple case where the KDC maintains a two-by-two array of secret keys as follows:

$K_{0,0}$	$K_{1,0}$
$K_{0,1}$	$K_{1,1}$

[0044] With this table, four possible receivers can be identified. They are the receivers identified by the combination of secret keys $K_{0,0}$ and $K_{1,0}$, $K_{0,0}$ and $K_{1,1}$, $K_{0,1}$ and $K_{1,0}$, and $K_{0,1}$ and $K_{1,1}$. A short-hand, way of representing these combinations is to represent them as a two digit number wherein the first digit represents the row assignment for column zero and the second digit represents the row assignment for column one. So, the four receivers can be identified as:

0,0 representing the receiver with key combination $K_{0,0}$ and $K_{1,0}$;

0,1 representing the receiver with key combination $K_{0,0}$ and $K_{1,1}$;

1,0 representing the receiver with key combination $K_{0,1}$ and $K_{1,0}$; and

1,1 representing the receiver with key combination $K_{0,1}$ and $K_{1,1}$.

[0045] In a first example, where all receivers are considered good, valid receivers, the KDC can simply send the content key (K_C) encrypted with the combination of secret keys for each receiver. That is, the KDC can send a structure to all receivers that contains:

K_C encrypted with $K_{0,0}$ and $K_{1,0}$ to receiver 0,0;

K_C encrypted with $K_{0,0}$ and $K_{1,1}$ to receiver 0,1;

K_C encrypted with $K_{0,1}$ and $K_{1,0}$ to receiver 1,0; and

K_C encrypted with $K_{0,1}$ and $K_{1,1}$ to receiver 1,1.

Sent along with the encrypted K_C should be an encrypted test pattern. This pattern should be one that is known to all receivers and is used to locate and verify the entries in the list of update keys that are intended for the individual receiver.

[0046] In a second, slightly more complex example, one of the four receivers may be considered to be "bad". That is, the receiver may have been hacked or perhaps a pay subscription has expired, or the receiver may be considered bad for other reasons. In any event, the content provider has determined that this receiver should no longer be authorized to decrypt content. To prevent this unauthorized receiver from receiving the content key K_C, intermediate keys are sent to all receivers such that a chain or progression of keys can be formed made up of good keys and bad keys in such a manner that good receivers can ultimately reach K_C while bad receivers cannot.

[0047] To illustrate, assume that receiver 1,0 has been found to be bad. The receiver can be identified at the KDC by its combination of secret keys. A table or list of keys to be sent to the receivers can be generated based on the array of secret keys stored on at the KDC. A series of intermediate update keys can then be sent to all receivers such that all receivers other than 1,0 receive keys that lead to a valid K_C while receiver 1,0 receives keys that result in an invalid combination.

[0048] This result can be achieved by building a list of update keys to be encrypted and sent to the receivers based on the table of secret keys stored at the KDC. Various methods of generating this table or list can be used. Likewise, this list may take on various forms while still accomplishing the basic goal of sending a series of encrypted intermediate update keys to all receivers. For the purpose of explanation, start with column 0. In this column, the number of possible bad receivers is indicated by row, column combinations present. In this example, one possible bad receiver is identified, 1,0. Therefore, two update keys should be generated by the KDC, One key (K_{U1}) for good receivers and one key (K_{U2}) for possibly bad receivers. Then, written into the table or list is:

Col. 0 row 0 gets K_{U1} encrypted with K_{U0}

Col. 0 row 1 gets K_{U2} encrypted with K_{U0}

[0049] Actually, K_{U1}, for example, is encrypted with the value obtained by combining Ku0, K0,0, and the table index as explained above. However, for clarity, this example simply uses K_{U0}.

[0050] Moving on to column 1, two possible keys have previously been used when building the list for column 0, K_{U1} and K_{U2}. So, there should be two list entries per row of column 1. Also, this is the last column of the table so receivers that are considered to be good receivers should be sent K_C. So, two keys can be sent, K_{U3} to bad receivers and K_C to good receivers. Col. 1, row 0 gets K_C or K_{U3} depending on column 0 and col. 1, row 1 gets K_C. The list then becomes:

Col. 1, row 0 gets K_C encrypted with K_{U1}

Col. 1, row 0 gets K_{U3} encrypted with K_{U2}

Col. 1, row 0 gets K_C encrypted with K_{U1}

Col. 1, row 0 gets K_C encrypted with K_{U2}

[0051] So, when the receivers parse the list of update keys and find the update keys based on a match of the test pattern, each receiver will have a series of three keys as follows:

Receiver 0,0 gets keys K_{U0}, K_{U1}, and K_C

Receiver 0,1 gets keys K_{U0}, K_{U1}, and K_C

Receiver 1,0 gets keys K_{U0}, K_{U2}, and K_{U3}

Receiver 1,1 gets keys K_{U0}, K_{U2}, and K_C

[0052] As a result, all good receivers end up with a combination of keys that result in a valid K_C while the bad receiver, receiver 1,0, ends up with a combination of keys that results in something other than a valid K_C.

[0053] In another, slightly more elaborate example, a table of 3 columns and 3 rows may be maintained by the KDC. For example:

$K_{0,0}$	$K_{1,0}$	$K_{2,0}$
$K_{0,1}$	$K_{1,1}$	$K_{2,1}$
$K_{0,2}$	$K_{1,2}$	$K_{2,2}$

[0054] This table provides 3^3 or 27 possible receivers that can be identified by various combinations of secret keys in the 3 columns. Using the short hand explained above, wherein the first digit represents the row assignment of the first column and so on, the receivers identified are:

0,0,0
0,0,1
0,0,2
0,1,0
0,1,1
0,1,2
0,2,0
0,2,1
0,2,2

1,0,0 (bad)
1,0,1
1,0,2
1,1,0
1,1,1
1,1,2
1,2,0
1,2,1
1,2,2

2,0,0
2,0,1 (bad)
2,0,2
2,1,0
2,1,1
2,1,2
2,2,0
2,2,1
2,2,2

[0055] As indicated, assume that receivers 1,0,0 and 2,0,1 are determined to be bad. All other receivers are considered good, valid receivers.

[0056] So, in generating a list of update keys for the receiver, the KDC may begin with column 0. Looking at this column, two possible bad receivers, 1,0,0 and 2,0,1, are identified along with numerous good receivers. Therefore, 3 update keys will be generated, K_{U1}, K_{U2}, and K_{U3}. Receivers identified with column 0 row 0, all of which are good receivers, will get K_{U1}. Receivers identified with column 0 row 1 or row 2 may be bad receivers and will get update keys K_{U2} or K_{U3}. So, the list of update keys becomes:

Col. 0 row 0 gets K_{U1} encrypted with K_{U0}

Col. 0 row 1 gets K_{U2} encrypted with K_{U0}

Col. 0 row 2 gets K_{U3} encrypted with K_{U0}

[0057] Moving on to column 1, there are 3 possible update keys generated based on column 0, so, each row of column 1 will have 3 entries. Additionally, there should be three possible update keys generated for column 1. One update key, K_{U4}, will be for the good receivers identified in rows 1 and 2, one update key, K_{U5}, will be for the first possible bad receivers path identified in row 0, and one update key, K_{U6}, will be the update key for the bad path from the previous column for row 2. So, the list becomes:

Col. 1 row 0 gets K_{U4} encrypted with K_{U1}

Col. 1 row 0 gets K_{U5} encrypted with K_{U2}

Col. 1 row 0 gets K_{U6} encrypted with K_{U3}

Col. 1 row 1 gets K_{U4} encrypted with K_{U1}

Col. 1 row 1 gets K_{U4} encrypted with K_{U2}

Col. 1 row 1 gets K_{U4} encrypted with K_{U3}

Col. 1 row 2 gets K_{U4} encrypted with K_{U1}

Col. 1 row 2 gets K_{U4} encrypted with K_{U2}

Col. 1 row 2 gets K_{U4} encrypted with K_{U3}

[0058] Moving on to column 2, there are three possible update keys coming into this column, K_{U4} K_{U5}, and K_{U6}. Therefore, each row of column 2 will have three entries in the list. Additionally, this is the last column of the table. So, good receivers will receive K_C while bad receives will receive something other than a valid K_C. In this example, bad receivers will be given K_U. The KDC then generates a list of update keys as follows:

Col. 2 row 0 gets K_C encrypted with K_{U4}

Col. 2 row 0 gets K_C encrypted with K_{U5}

Col. 2 row 0 gets K_{U7} encrypted with K_{U6}

Col. 2 row 1 gets K_C encrypted with K_{U4}

Col. 2 row 1 gets K_{U7} encrypted with K_{U5}

Col. 2 row 1 gets K_C encrypted with K_{U6}

Col. 2 row 2 gets K_C encrypted with K_{U4}

Col. 2 row 2 gets K_C encrypted with K_{U5}

Col. 2 row 2 gets K_C encrypted with K_{U6}

[0059] So, when the receivers parse the list of update keys and find the update keys based on a match of the test pattern, each receiver will have a series of three keys as follows:

Receiver 0,0,0 gets keys K_{U0}, K_{U1}, K_{U4} and K_C

Receiver 0,0,1 gets keys K_{U0}, K_{U1}, K_{U4} and K_C

Receiver 0,0,2 gets keys K_{U0}, K_{U1}, K_{U4} and K_C

Receiver 0,1,0 gets keys K_{U0}, K_{U1}, K_{U4} and K_C

Receiver 0,1,1 gets keys K_{U0}, K_{U1}, K_{U4} and K_C

Receiver 0,1,2 gets keys K_{U0}, K_{U1}, K_{U4} and K_C

Receiver 0,2,0 gets keys K_{U0}, K_{U1}, K_{U4} and K_C

Receiver 0,2,1 gets keys K_{U0}, K_{U1}, K_{U4} and K_C

Receiver 0,2,2 gets keys K_{U0}, K_{U1}, K_{U4} and K_C

Receiver 1,0,0 gets keys K_{U0}, K_{U2}, K_{U6} and K_{U7} (bad)

Receiver 1,0,1 gets keys K_{U0}, K_{U2}, K_{U6} and K_C

Receiver 1,0,2 gets keys K_{U0}, K_{U2}, K_{U6} and K_C

Receiver 1,1,0 gets keys K_{U0}, K_{U2}, K_{U5} and K_C

Receiver 1,1,1 gets keys K_{U0}, K_{U2}, K_{U5} and K_C

Receiver 1,1,2 gets keys K_{U0}, K_{U2}, K_{U5} and K_C

Receiver 1,2,0 gets keys K_{U0}, K_{U2}, K_{U4} and K_C

Receiver 1,2,1 gets keys K_{U0}, K_{U2}, K_{U4} and K_C

Receiver 1,2,2 gets keys K_{U0}, K_{U2}, K_{U4} and K_C

Receiver 2,0,0 gets keys K_{U0}, K_{U3}, K_{U5} and K_C

Receiver 2,0,1 gets keys K_{U0}, K_{U3}, K_{U5} and K_{U7} (bad)

Receiver 2,0,2 gets keys K_{U0}, K_{U3}, K_{U5} and K_C

Receiver 2,1,0 gets keys K_{U0}, K_{U3}, K_{U4} and K_C

Receiver 2,1,1 gets keys K_{U0}, K_{U3}, K_{U4} and K_C

Receiver 2,1,2 gets keys K_{U0}, K_{U3}, K_{U4} and K_C

Receiver 2,2,0 gets keys K_{U0}, K_{U3}, K_{U4} and K_C

Receiver 2,2,1 gets keys K_{U0}, K_{U3}, K_{U4} and K_C

Receiver 2,2,2 gets keys K_{U0}, K_{U3}, K_{U4} and K_C

[0060] As a result, good receivers end up with a chain of update keys resulting in a valid content key while bad receivers 1,0,0 and 2,0,1 end up with an invalid content key.